

# Licensing support by distro generators and what remains to be done

Jan Altenberg

Open Source Automation Development Lab (OSADL) eG

# Building an embedded Linux system

- Custom build
- Distribution generator
  - PTXDist
  - Buildroot
  - Yocto project
- Based on Distribution (e.g. Debian)
  - ELBE (Embedded Linux Build Environment)
  - ISAR (Integration System for Automated Root filesystem generation)
  - Debos (Debian OS images builder)

# Building an embedded Linux system

- Custom build
- Distribution generator
  - PTXDist
  - Buildroot
  - **Yocto project**
- Based on Distribution (e.g. Debian)
  - **ELBE (Embedded Linux Build Environment)**
  - **ISAR (Integration System for Automated Root filesystem generation)**
  - Debos (Debian OS images builder)

# License obligations: Overview

- Information obligations
- Disclosure obligations
- Licensing obligations
- Additional obligations

# License obligations: Overview

- Information obligations
- Disclosure obligations
- Licensing obligations
- Additional obligations

## Provide:

- License texts
- Warranty disclaimers
- Modification notices
- Acknowledgements

# License obligations: Overview

- Information obligations
- Disclosure obligations
- Licensing obligations
- Additional obligations

## Provide:

- Complete corresponding source code
- Including modifications
- Build and installation information

# License obligations: Overview

- Information obligations
- Disclosure obligations
- Licensing obligations
- Additional obligations

**Licenses with a copyleft clause (e.g. GPL) require derivative works (i.e. created by modifying or linking) to be licensed under the original license when distributing.**

# License obligations: Overview

- Information obligations
- Disclosure obligations
- Licensing obligations
- Additional obligations

**Adapting company documents to account for Open Source licenses, e.g. Terms and Conditions must give precedence to Open Source licenses.**



# License obligations: Overview

- Information obligations
- Disclosure obligations
- Licensing obligations
- Additional obligations

**E.g. the Lesser GNU General Public License (LGPL) requires to give permission to modify and reengineer linked proprietary software.**

# License obligations and distro generators

- Information obligations
- Disclosure obligations
- Licensing obligations
- Additional obligations

**BUT HOW CAN A DISTRO GENERATOR SUPPORT?!**

# License obligations and distro generators

- Information obligations
- Disclosure obligations
- Licensing obligations
- ~~Additional obligations~~

**BUT HOW CAN A DISTRO GENERATOR SUPPORT?!**

# License obligations and distro generators

- Information obligations
- Disclosure obligations
- Licensing obligations
- ~~Additional obligations~~

**BUT HOW CAN A DISTRO GENERATOR SUPPORT?!  
MANUAL / ADDITIONAL EFFORT IS ALWAYS NEEDED**

# Yocto: Overview

- Yocto is an umbrella project, providing tools and methods for building custom Linux-based systems.
- Yocto provides a reference distribution called “Poky”.
- <https://git.yoctoproject.org/poky>

# Yocto: Information obligations

```
./tmp/deploy/licenses/qemux86_64/core-image-  
minimal-qemux86-64.rootfs/
```

```
|— image_license.manifest  
|— license.manifest  
|— package.manifest
```

# Yocto: Information obligations

```
./tmp/deploy/licenses/qemux86_64/core-image-  
minimal-qemux86-64.rootfs/
```

- └─ image\_license.manifest
- └─ license.manifest
- └─ package.manifest

**Components needed to  
boot the device, not  
being part of the  
rootfilesystem**

# Yocto: Information obligations

```
./tmp/deploy/licenses/qemux86_64/core-image-  
minimal-qemux86-64.rootfs/
```

```
|— image_license.manifest  
|— license.manifest  
|— package.manifest
```

**License information for  
all packages**



# Yocto: Information obligations

```
./tmp/deploy/licenses/qemux86_64/core-image-  
minimal-qemux86-64.rootfs/
```

```
|— image_license.manifest  
|— license.manifest  
|— package.manifest
```

[...]

```
PACKAGE NAME: kernel-image  
PACKAGE VERSION: 6.6.35+git  
RECIPE NAME: linux-yocto  
LICENSE: GPL-2.0-only
```

[...]

# Yocto: Information obligations

```
./tmp/deploy/licenses/qemux86_64/core-image-  
minimal-qemux86-64.rootfs/
```

- └─ image\_license.manifest
- └─ license.manifest
- └─ package.manifest

**List of the packages  
installed into the image**

# Yocto: Information obligations

- Poky generates a **license manifest** when building an image. The license manifest and the licenses can be copied into the generated rootfilesystem:

```
build/conf/local.conf:  
[...]  
COPY_LIC_MANIFEST = "1"  
COPY_LIC_DIRS = "1"
```

# Yocto: Information obligations

rootfs/usr/share/common-licenses/busybox

- |— generic\_bzip2-1.0.4 -> ../generic\_bzip2-1.0.4
- |— generic\_GPL-2.0-only -> ../generic\_GPL-2.0-only
- |— LICENSE.0
- |— LICENSE.1
- |— recipeinfo

# Where does the data come from?

License manifest:

PACKAGE NAME: kernel-image  
PACKAGE VERSION: 6.6.35+git  
RECIPE NAME: linux-yocto  
LICENSE: GPL-2.0-only

Recipe:

SUMMARY = "Linux kernel"  
SECTION = "kernel"  
LICENSE = "GPL-2.0-only"  
HOMEPAGE = "https://www.yoctoproject.org/"

# Where does the data come from?

License manifest:

PACKAGE NAME: kernel-image

PACKAGE VERSION: 6.6.35+git

RECIPE NAME: linux-yocto

**LICENSE: GPL-2.0-only**

Recipe:

SUMMARY = "Linux kernel"

SECTION = "kernel"

**LICENSE = "GPL-2.0-only"**

HOMEPAGE = "https://www.yoctoproject.org/"

# Where does the data come from?

License manifest:

PACKAGE NAME: kernel-image

PACKAGE VERSION: 6.6.35+git

RECIPE NAME: linux-yocto

**LICENSE: GPL-2.0-only**

Recipe:

SUMMARY = "Linux kernel"

SECTION = "kernel"

**LICENSE = "GPL-2.0-only"**

HOMEPAGE = "https://www.yoctoproject.org/"

Is this enough?

# License expressions - Linux Kernel 6.6.35

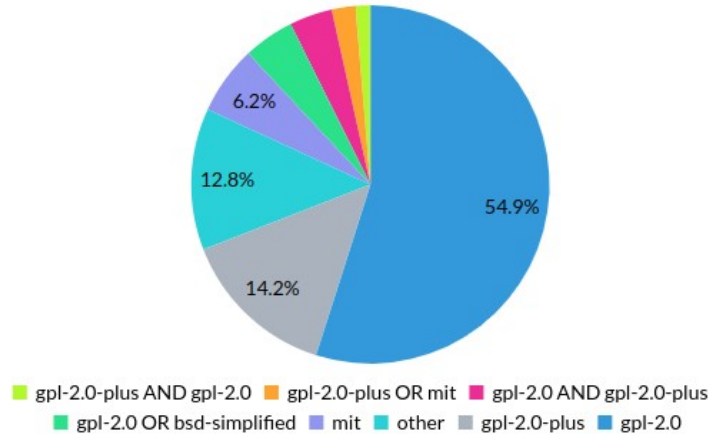
102

Total licenses

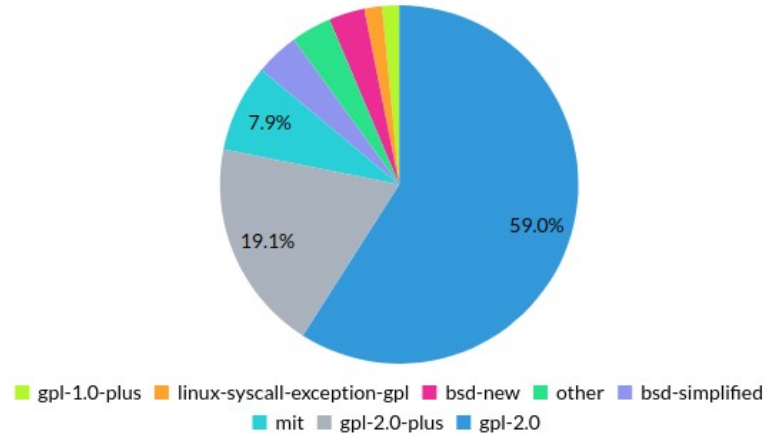
74271

Total files with licenses

License expression



License keys





# Yocto: Create SPDX files

build/conf/local.conf:

```
INHERIT += "create-spdx"
```

**This is set by default in recent versions.**

# Yocto: Create SPDX files

build/conf/local.conf:

```
INHERIT += "create-spdx"  
SPDX_PRETTY = "1"  
SPDX_INCLUDE_SOURCES = "1"  
SPDX_ARCHIVE_SOURCES = "1"  
SPDX_ARCHIVE_PACKAGED = "1"
```

# Yocto: Create SPDX files

build/conf/local.conf:

```
INHERIT += "create-spdx"  
SPDX_PRETTY = "1"  
SPDX_INCLUDE_SOURCES = "1"  
SPDX_ARCHIVE_SOURCES = "1"  
SPDX_ARCHIVE_PACKAGED = "1"
```

**Human readable output  
(default output is in one  
single line)**

# Yocto: Create SPDX files

build/conf/local.conf:

```
INHERIT += "create-spdx"  
SPDX_PRETTY = "1"  
SPDX_INCLUDE_SOURCES = "1"  
SPDX_ARCHIVE_SOURCES = "1"  
SPDX_ARCHIVE_PACKAGED = "1"
```

**Adds a description of  
the source files used to  
build the host tools and  
the packages for the  
target**

# Yocto: Create SPDX files

build/conf/local.conf:

```
INHERIT += "create-spdx"  
SPDX_PRETTY = "1"  
SPDX_INCLUDE_SOURCES = "1"  
SPDX_ARCHIVE_SOURCES = "1"  
SPDX_ARCHIVE_PACKAGED = "1"
```

**Archives the sources of  
the packages installed  
on the target (only  
works in combination  
with  
\_INCLUDE\_SOURCES)**

# Yocto: Create SPDX files

build/conf/local.conf:

```
INHERIT += "create-spdx"  
SPDX_PRETTY = "1"  
SPDX_INCLUDE_SOURCES = "1"  
SPDX_ARCHIVE_SOURCES = "1"  
SPDX_ARCHIVE_PACKAGED = "1"
```

**Archives the binaries of  
the packages installed  
on the target**

# Yocto: Create SPDX files

`tmp/deploy/images/qemux86-64/core-image-minimal-qemux86-64.rootfs-DATE.spdx.tar.zst:`

`core-image-minimal-qemux86-64.rootfs-DATE.spdx.json`

`base-files.spdx.json`

`recipe-base-files.spdx.json`

`runtime-base-files.spdx.json`

`[...]`

`index.json`

# Yocto: Create SPDX files

core-image-minimal-qemux86-64.rootfs-DATE.spdx.json



index.json



base-files.spdx.json recipe-base-files.spdx.json ...



# Yocto: Create SPDX files

core-image-minimal-qemux86-64.rootfs-DATE.spdx.json



index.json



base-files.spdx.json recipe-base-files.spdx.json ...

# Yocto: Create SPDX files

```
[...]  
"homepage": "https://www.busybox.net",  
"licenseConcluded": "NOASSERTION",  
"licenseDeclared": "GPL-2.0-only AND LicenseRef-bzip2-1.0.4",  
"licenseInfoFromFiles": [  
  "NOASSERTION"  
],  
"name": "busybox",  
"packageFileName": "recipe-busybox.tar.zst",  
"packageVerificationCode": {  
  "packageVerificationCodeValue":  
    "1def1818b2f758a279ecb5a14fdcd11b671a902d"  
},
```

# Yocto: Create SPDX files

```
[...]  
],  
"copyrightText": "NOASSERTION",  
"fileName": "busybox-1.36.1/applets/applets.c",  
"fileTypes": [  
    "SOURCE"  
],  
"licenseConcluded": "NOASSERTION",  
"licenseInfoInFiles": [  
    "NOASSERTION"  
]  
},
```

# Yocto: Create SPDX files

```
[...]  
],  
"copyrightText": "NOASSERTION",  
"fileName": "busybox-1.36.1/applets/applets.c",  
"fileTypes": [  
    "SOURCE"  
],  
"licenseConcluded": "NOASSERTION",  
"licenseInfoInFiles": [  
    "NOASSERTION"  
]  
},
```

CREATE-SPDX ONLY CATCHES SPDX-LICENSE-IDENTIFIERS

# Yocto: Create SPDX files

```
[...]  
],  
"copyrightText": "NOASSERTION",  
"fileName": "busybox-1.36.1/applets/applets.c",  
"fileTypes": [  
    "SOURCE"  
],  
"licenseConcluded": "NOASSERTION",  
"licenseInfoInFiles": [  
    "NOASSERTION"  
]  
},
```

**ONLY SCANNING RESULTS, NO MANUAL CONCLUSION**

# Yocto: meta-spxscanner

- <https://git.yoctoproject.org/meta-spxscanner/>
- Provides classes for the integration of different scanners:
  - blackduck-upload: Blackduck upload using Synopsis Detect
  - bom: Integration of the SBOM Multitool
  - fossology-python and fossology-rest: FOSSology integration
  - scancode-tk: Integration of the scancode toolkit

# Yocto: meta-spxscanner

- <https://git.yoctoproject.org/meta-spxscanner/>
- Provides classes for the integration of different scanners:
  - blackduck-upload: Blackduck upload using Synopsis Detect
  - bom: Integration of the SBOM Multitool
  - fossology-python and fossology-rest: FOSSology integration
  - **scancode-tk: Integration of the scancode toolkit**

# Yocto: meta-spxscanner

```
# bblayers.conf:
```

```
BBLAYERS ?= " \  
    /path/to/oe-core/meta \  
    [...]  
    /path/to/layer/meta-spxscanner "
```



# Yocto: meta-spxscanner

```
# local.conf:
```

```
INHERIT += "cve-check"
```

```
CVE_CHECK_FORMAT_JSON = "0"
```

```
INHERIT += "scancode-tk"
```

# Yocto: meta-spxscanner

tmp/deploy/spx/busybox-1.36.1.spx:

PackageCopyrightText: <text>

Copyright (c) 1980 The Regents of the University of California

Copyright (c) 1983, 1993 The Regents of the University of California

Copyright (c) 1983,1991 The Regents of the University of California

Copyright (c) 1984, 1989, 1990, 2000, 2001, 2002, 2003, 2004 Free  
Software Foundation, Inc.

[...]

# Yocto: meta-spxscanner

[...]

FileName: ./work/busybox-1.36.1/applets/applets.c

SPDXID: SPDXRef-busybox

FileChecksum: SHA1: 2e6d03880c120566f1460c0e11cf9555d092eeb4

LicenseConcluded: NOASSERTION

LicenseInfoInFile: GPL-2.0-or-later

[...]

# Yocto: meta-spxscanner

[...]

FileName: ./work/busybox-1.36.1/applets/applets.c

SPDXID: SPDXRef-busybox

FileChecksum: SHA1: 2e6d03880c120566f1460c0e11cf9555d092eeb4

LicenseConcluded: NOASSERTION

LicenseInfoInFile: GPL-2.0-or-later

[...]



**Scanner finding**

# Yocto: meta-spxscanner

[...]

FileName: ./work/busybox-1.36.1/applets/applets.c

SPDXID: SPDXRef-busybox

FileChecksum: SHA1: 2e6d03880c120566f1460c0e11cf9555d092eeb4

LicenseConcluded: NOASSERTION

LicenseInfoInFile: GPL-2.0-or-later

[...]



**No manual conclusion**

# Yocto: meta-ossebot

- Integrates **OSsebot** in the bitbake build process.

# Yocto: meta-ossebot

- Integrates **OSsebot** in the bitbake build process.

<https://www.ossebot.org/>

# Yocto: meta-osselet

- <https://github.com/iris-GmbH/meta-osselet>
- For all packages going into the target image, the OSSelot database is checked for the closest available version and the data is taken from the repository.
- The checksums of all source files are compared against the curated data in the OSSelot database and the results are documented in a JSON file.



# Yocto: meta-osselet

```
# bblayers.conf:
```

```
BBLAYERS ?= " \  
    /path/to/oe-core/meta \  
    [...]  
    /path/to/layer/meta-osselet "
```

# Yocto: meta-osselet

```
# local.conf:
```

```
[...]
```

```
INHERIT += "osselet"
```

```
[...]
```

```
$ bitbake core-image-minimal --runonly=populate_osselet
```

# Yocto: meta-osselet

```
$ $ ls tmp/deploy/osselet/busybox/  
busybox-1.36.1-meta.json  version-1.36.1
```

# Yocto: meta-osselet

```
$ $ ls tmp/deploy/osselet/busybox/  
busybox-1.36.1-meta.json  version-1.36.1
```

**Data taken from the  
OSSelot repository**

# Yocto: meta-osselet

```
$ $ ls tmp/deploy/osselet/busybox/  
busybox-1.36.1-meta.json  version-1.36.1
```

**Metadata: E.g.  
checksum matches /  
mismatches**

# Yocto: meta-osselot

WARNING: openssl-3.2.2-r0 do\_osselot\_populate\_workdir: openssl/3.2.2 not available in osselot database. Using version openssl/3.2.0

# Yocto: meta-osselet

WARNING: openssl-3.2.2-r0 do\_osselet\_populate\_workdir: openssl/3.2.2 not available in osselet database. Using version openssl/3.2.0

```
# tmp/deploy/osselet/openssl/openssl-3.2.2-meta.json:
```

```
"spdx_checksum_data_mismatch": [  
    "VERSION.dat",  
    "providers/implementations/ciphers/cipher_aes_hw_s390x.inc",  
    "providers/implementations/ciphers/cipher_aes_gcm_hw.c",
```

# Yocto: meta-osselet

```
WARNING: openssl-3.2.2-r0 do_osselet_populate_workdir: openssl/3.2.2 not  
available in osselet database. Using version openssl/3.2.0
```

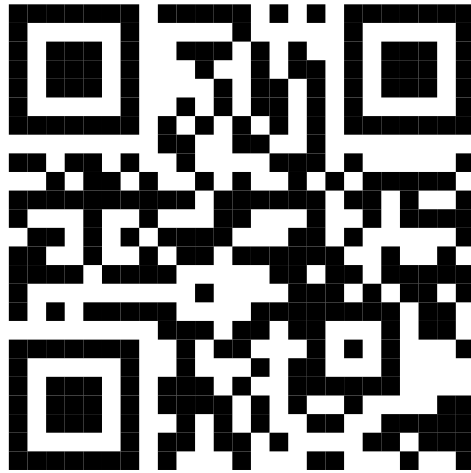
```
# tmp/deploy/osselet/openssl/openssl-3.2.2-meta.json:
```

```
"spdx_checksum_data_mismatch": [  
    "VERSION.dat",  
    "providers/implementations/ciphers/cipher_aes_hw_s390x.inc",  
    "providers/implementations/ciphers/cipher_aes_gcm_hw.c",  
    [...]  
"spdx_checksum_equivalence_data_match": [],  
"spdx_checksum_data_match": [  
    "providers/implementations/ciphers/cipher_aes_xts.h",  
    "providers/implementations/ciphers/cipher_rc2.h",
```



# Yocto: meta-osselet

- Webinar on September 25, 2024 (2pm to 4pm CEST)  
Integrating OSSelot curation data into OpenEmbedded
- Free registration on <https://www.osadl.org/?id=3420>



# Yocto: Disclosure obligations

build/conf/local.conf:

[...]

INHERIT += "archiver"

ARCHIVER\_MODE = "original"

# Yocto: Disclosure obligations

build/conf/local.conf:

```
[...]
```

```
INHERIT += "archiver"
```

```
ARCHIVER_MODE = "original"
```

**"original": Original sourcecode**

# Yocto: Disclosure obligations

build/conf/local.conf:

[...]

INHERIT += "archiver"

ARCHIVER\_MODE = **"configured"**

"original": Original sourcecode

**"configured": Patched and configured sourcecode**

# Yocto: Disclosure obligations

build/conf/local.conf:

[...]

INHERIT += "archiver"

ARCHIVER\_MODE = **"patched"**

"original": Original sourcecode

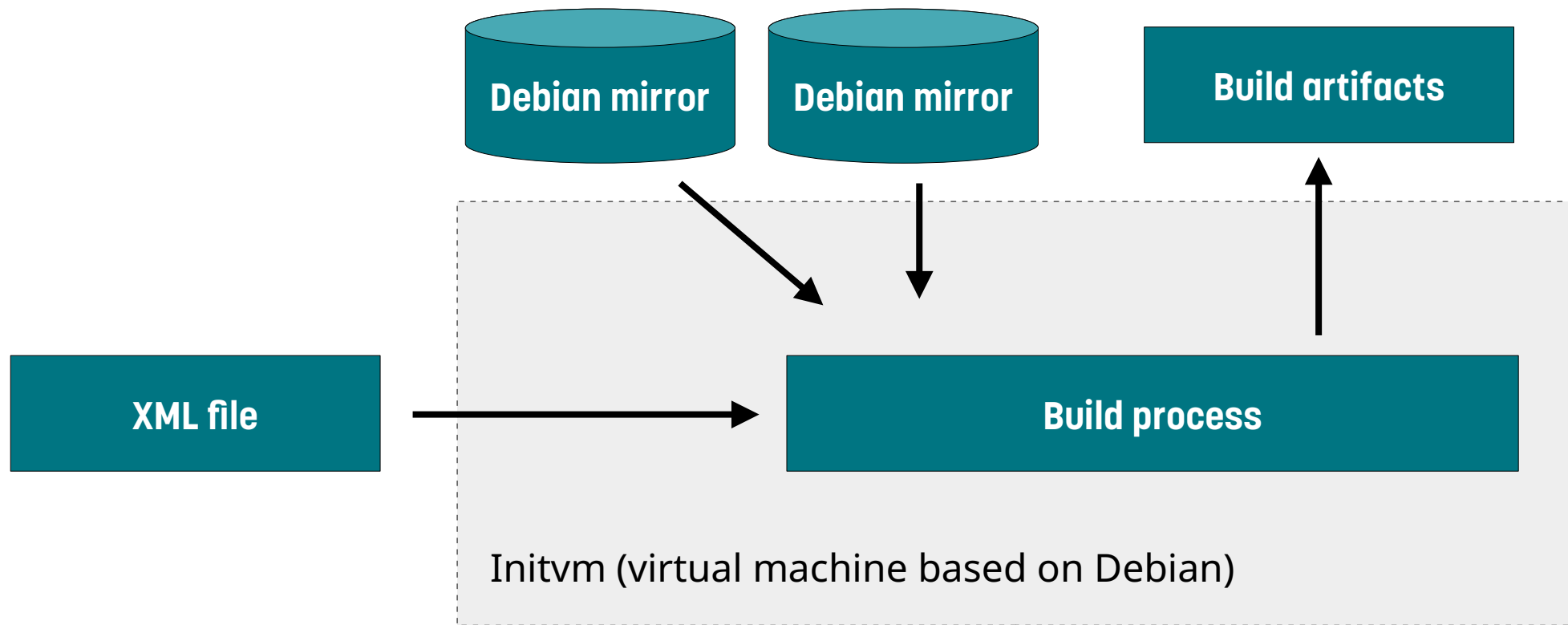
"configured": Patched and configured sourcecode

**"patched": Patched sourcecode**

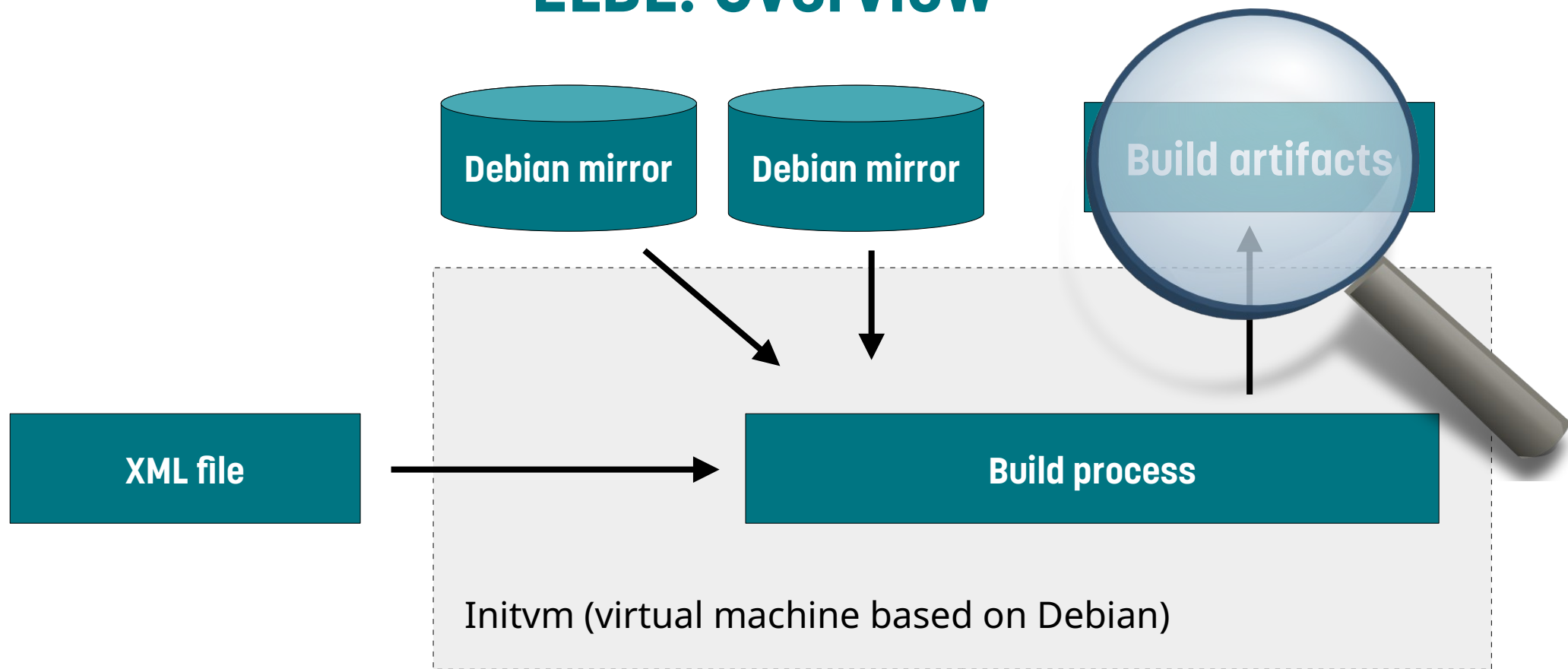
# ELBE: Overview

- <https://elbe-rfs.org/>
- ELBE is a tool for building rootfilesystems and images based on Debian.
- The target image is described in an XML file.

# ELBE: Overview



# ELBE: Overview





# ELBE: Fulfilling license obligations

elbe-build-20240127-105058/

- bin-cdrom.iso
- licence-chroot.txt
- licence-chroot.xml
- licence-target.txt
- licence-target.xml
- source.xml
- src-cdrom-added.iso
- src-cdrom-main.iso
- src-cdrom-target.iso

**ISO image with all  
installed packages**

# ELBE: Fulfilling license obligations

elbe-build-20240127-105058/

- bin-cdrom.iso
- licence-chroot.txt
- licence-chroot.xml
- **licence-target.txt**
- **licence-target.xml**
- source.xml
- src-cdrom-added.iso
- src-cdrom-main.iso
- src-cdrom-target.iso

**Licenses and copyright  
information for the target  
packages (in TXT and  
XML format)**

# ELBE: Fulfilling license obligations

elbe-build-20240127-105058/

- bin-cdrom.iso
- licence-chroot.txt
- licence-chroot.xml
- licence-target.txt
- licence-target.xml
- **source.xml**
- src-cdrom-added.iso
- src-cdrom-main.iso
- src-cdrom-target.iso

**Project XML with all  
package versions being  
used for the build. Can be  
used as input for  
reproducible builds.**

# ELBE: Fulfilling license obligations

elbe-build-20240127-105058/

- bin-cdrom.iso
- licence-chroot.txt
- licence-chroot.xml
- licence-target.txt
- licence-target.xml
- source.xml
- src-cdrom-added.iso
- src-cdrom-main.iso
- src-cdrom-target.iso

**ISO images with the  
Debian source packages  
for the build**

# ELBE: Information obligations

licence-target.txt:

apt:

=====

Apt is copyright 1997, 1998, 1999 Jason Gunthorpe and others.

Apt is currently developed by APT Development Team <deity@lists.debian.org>.

License: GPLv2+

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or [...]

# Where does the data come from?

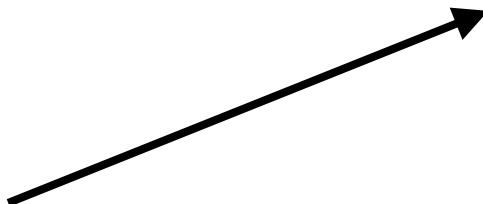
elbe-build-20240127-105058/

- bin-cdrom.iso
- licence-chroot.txt
- licence-chroot.xml
- licence-target.txt
- licence-target.xml
- source.xml
- src-cdrom-added.iso
- src-cdrom-main.iso
- src-cdrom-target.iso**

apt-1.8.2.3/COPYING



pool/target/a/apt/apt\_1.8.2.3.tar.xz



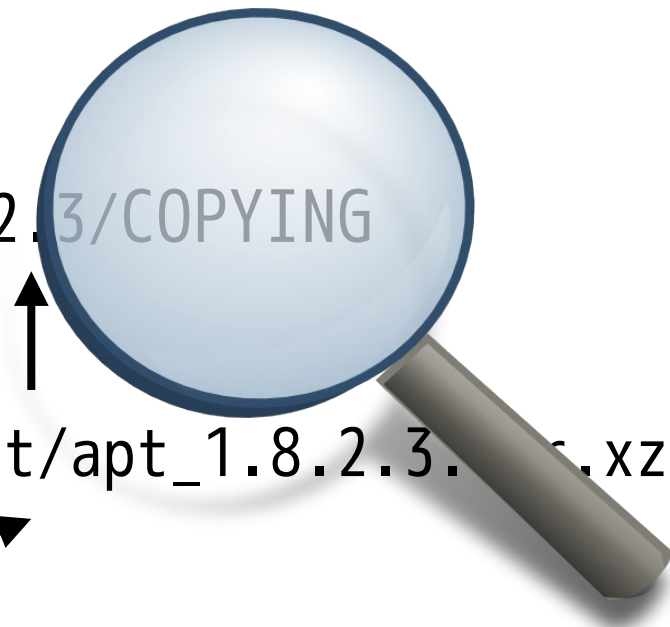
# Where does the data come from?

elbe-build-20240127-105058/

- bin-cdrom.iso
- licence-chroot.txt
- licence-chroot.xml
- licence-target.txt
- licence-target.xml
- source.xml
- src-cdrom-added.iso
- src-cdrom-main.iso
- src-cdrom-target.iso**

apt-1.8.2.3/COPYING

pool/target/a/apt/apt\_1.8.2.3...xz



# Where does the data come from?

Apt is copyright 1997, 1998, 1999 Jason Gunthorpe and others.  
Apt is currently developed by APT Development Team  
<deity@lists.debian.org>.

License: GPLv2+

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
[...]



# Where does the data come from?

Apt is copyright 1997, 1998, 1999 Jason Gunthorpe and others.  
Apt is currently developed by APT Development team  
<deity@lists.debian.org>.

License: GPLv2+

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
[...]

**META INFORMATION FROM THE DEBIAN PACKAGES IS TAKEN.  
QUALITY DIFFERS AND IS USUALLY NOT COMPLETE.**

# ELBE: SPDX

```
$ elbe parselicense  
--tvout=out.spdx \  
--mapping=map.xml \  
licence-target.xml
```

# ELBE: SPDX

```
$ elbe parselicense  
--tvout=out.spdx \  
--mapping=map.xml \  
licence-target.xml
```

```
<mapping name='GPL-1+'>GPL-1.0-or-later</mapping>  
<mapping name='GPL-2'>GPL-2.0-only</mapping>  
<mapping name='GPL-2.0'>GPL-2.0-only</mapping>  
<mapping name='GPL-2+'>GPL-2.0-or-later</mapping>  
<mapping name='GPLv2+'>GPL-2.0-or-later</mapping>  
[...]
```

# ISAR: Overview

- <https://github.com/ilbers/isar>
- Set of scripts for building rootfilesystems and images based on Debian
- Uses bitbake
- Since ISAR is based on bitbake, there are quite some similarities to Yocto.

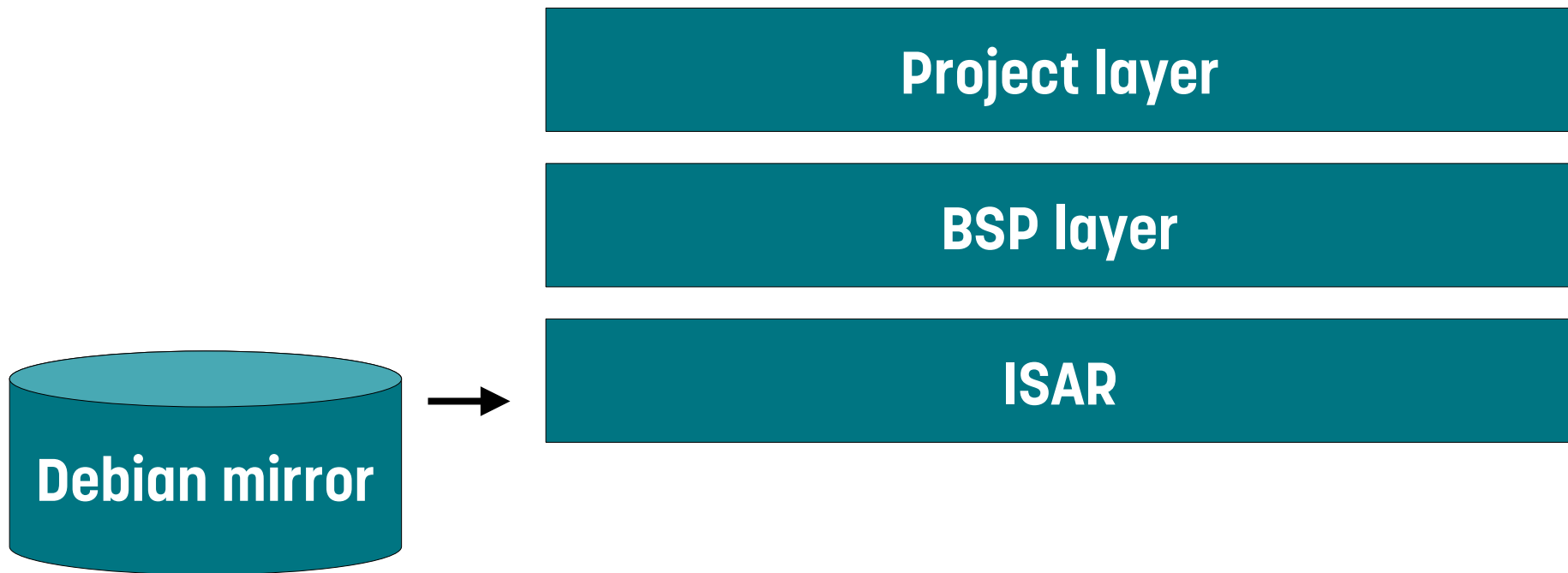
# ISAR: Overview

**Project layer**

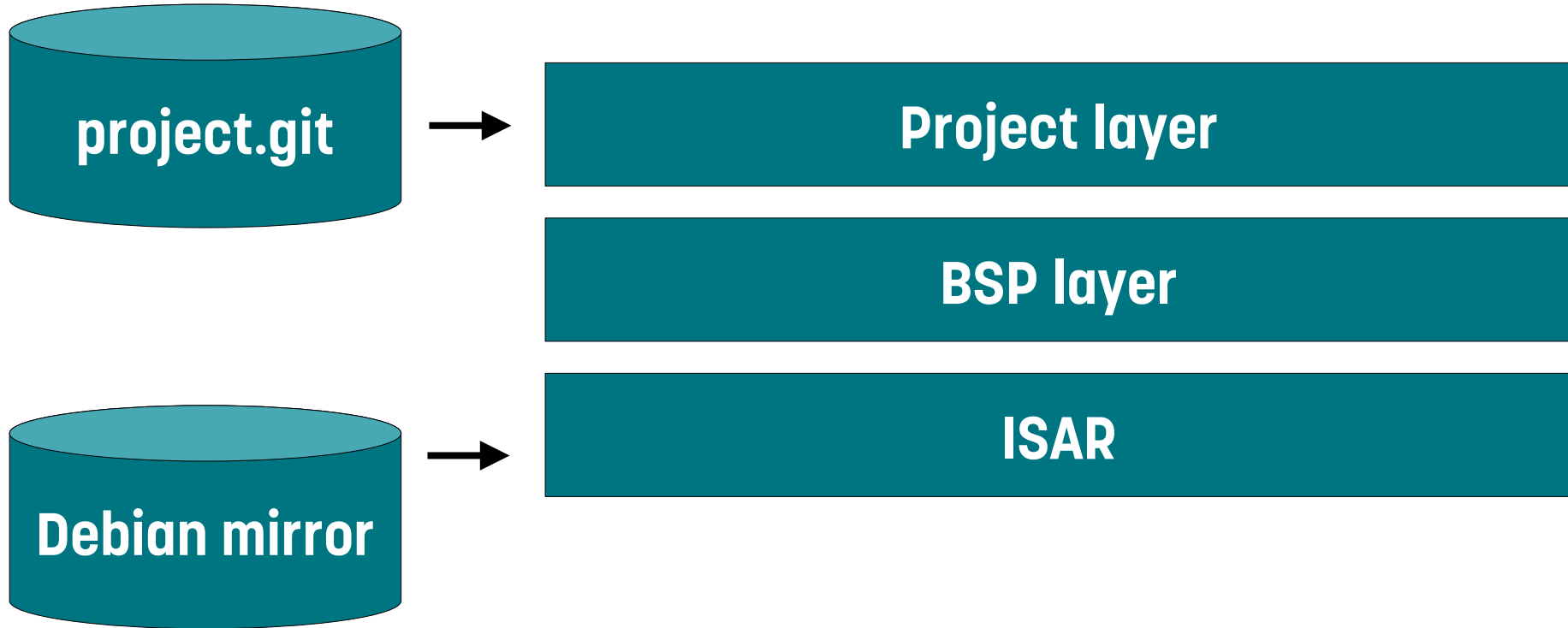
**BSP layer**

**ISAR**

# ISAR: Overview



# ISAR: Overview



# ISAR: Information obligations

- Similar to Yocto a manifest file is created in: `tmp/deploy/images/`
- ISAR also relies on the meta information.
- Therefore, license and copyright information is not complete.



# Where does the data come from?

**CUSTOM RECIPES OR DEBIAN PACKAGE INFORMATION  
(SIMILAR TO YOCTO / ELBE)**

# ISAR: Disclosure obligations

- Setting `BASE_REPO_FEATURES = "cache-deb-src"` in the `local.conf` file will download the Debian source packages to: `build/downloads/deb-src/`
- Custom packages coming from ISAR layers are placed in: `build/tmp/work/`

# Summary (Yocto / information obligations)

|                      |   |
|----------------------|---|
| Licenses             | Incomplete (can be extended by additional layers) |
| Copyright notices    | No (can be added with additional layers)          |
| Warranty disclaimer  | No  |
| Acknowledgments      | No  |
| Modification notices | Modified source code                              |

# Summary (Yocto / disclosure obligations)

|                    |   |
|--------------------|---|
| Build instructions | Yes (in recipes)  |
| Patches            | Yes (available in the layers or in the code with archiver mode properly configured) |

# Summary (ELBE / information obligations)

|                      |                      |
|----------------------|----------------------|
| Licenses             | Incomplete           |
| Copyright notices    | Incomplete           |
| Warranty disclaimer  | No                   |
| Acknowledgments      | No                   |
| Modification notices | Modified source code |

# Summary (ELBE / disclosure obligations)

|                    |                                 |
|--------------------|---------------------------------|
| Build instructions | Yes (in Debian source packages) |
| Patches            | Yes (in Debian source packages) |

# Summary (ISAR / information obligations)

|                      |                      |
|----------------------|----------------------|
| Licenses             | Incomplete           |
| Copyright notices    | Incomplete           |
| Warranty disclaimer  | No                   |
| Acknowledgments      | No                   |
| Modification notices | Modified source code |

# Summary (ISAR / disclosure obligations)

|                    |   |
|--------------------|---|
| Build instructions | Yes (in recipes / Debian source packages) |
| Patches            | Yes (in recipes / Debian source packages) |



# Summary

- Distro generators can support with fulfilling license obligations.
- BUT: Additional effort is always needed!
- Distro generators offer strong support for fulfilling disclosure obligations.
- Distro generators rely on given meta information for fulfilling information obligations:
  - Information is incomplete.
  - Source code has to be scanned and information has to be extracted.
  - The **OSSelot** project provides a community for sharing and reusing such data.